

AFRL-IF-RS-TR-2003-197
Final Technical Report
August 2003



PERPETUAL TESTING

University of Massachusetts

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. E097

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-197 has been reviewed and is approved for publication.

APPROVED: /s/
DEBORAH A. CERINO
Project Engineer

FOR THE DIRECTOR: /s/
JAMES A. COLLINS, Acting Chief
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 074-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)			2. REPORT DATE AUGUST 2003		3. REPORT TYPE AND DATES COVERED Final Feb 97 – Dec 01	
4. TITLE AND SUBTITLE PERPETUAL TESTING			5. FUNDING NUMBERS C - F30602-97-2-0032 PE - 63760E PR - E097 TA - 01 WU - 01			
6. AUTHOR(S) Leon J. Osterwell and Lori A. Clarke			8. PERFORMING ORGANIZATION REPORT NUMBER N/A			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Massachusetts 140 Governors Drive Amherst Massachusetts 01003-4610						
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFTB 525 Brooks Road Rome New York 13441-4505			10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2003-197			
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Deborah A. Cerino/IFTB/(315) 330-1445/ Deborah.Cerino@rl.af.mil						
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 Words) The University of Massachusetts participated with Purdue University and the University of California at Irvine (under separate contractual efforts) in a teamed research effort to advance the state-of-the-art in software testing. The University of Massachusetts has collaborated with its research partners on developing improved data flow analysis capabilities and on the development of integrated testing strategies that synergistically combine static and dynamic approaches. This effort is part of the DARPA Evolutionary Design of Complex Software (EDCS) Program. The EDCS project made important progress along numerous fronts. Research activity was centered on: Finite State Verification Software Process Technology Self-Adaptive Software Distributed Object Technology						
In addition, there was considerable activity aimed at dissemination, publication, and student training.						
14. SUBJECT TERMS Software Testing, Static and Dynamic Analysis, Software Process Language						15. NUMBER OF PAGES 16
						16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL			

Table of Contents

RESEARCH ACTIVITIES:.....	1
Finite state verification:	1
Software Process Technology:.....	2
Language Features	3
Self Adaptive Software:.....	4
Distributed Object Technology:.....	4
DISSEMINATION, PUBLICATION, AND STUDENT TRAINING:	4
Publications:.....	4

The University of Massachusetts participated with Purdue University and the University of California at Irvine (under separate contractual efforts) in a teamed research effort to advance the state-of-the-art in software testing. The University of Massachusetts has collaborated with its research partners on developing improved data flow analysis capabilities and on the development of integrated testing strategies that synergistically combine static and dynamic approaches. This effort is part of the DARPA Evolutionary Design of Complex Software (EDCS) Program.

The EDCS project made important progress along numerous fronts. Research activity was centered on work on:

- **Finite state verification**
- **Software process technology**
- **Self adaptive software**
- **Distributed object technology**

In addition, there was considerable activity aimed at dissemination, publication, and student training.

We first briefly summarize the research accomplishments, and then list the many dissemination activities wherein can be found details of our accomplishments.

RESEARCH ACTIVITIES:

Finite state verification:

On this project we made important progress in our work with finite state verification. Although we continued our work on applying our Flow Analysis for Verification of Systems (FLAVERS) data flow analysis system to Ada software, towards the end of the project, our emphasis and focus shifted to applying FLAVERS to Java software.

FLAVERS is a flexible, powerful system for automatically guaranteeing the absence or detecting the presence of a wide range of user-specified properties or behaviors in both sequential and concurrent systems. FLAVERS complements traditional testing approaches, which only demonstrate the presence or absence of errors for the specific test cases that have been executed. It also complements formal verification methods, which employ more comprehensive analysis, but require extensive expertise on the part of the user. FLAVERS is built using the FLAVERS toolkit, a flexible architecture and component library that facilitates the development of FLAVERS analyzers. The toolkit allows FLAVERS to be extended with additional specialized data flow analyzers and to be applied to a wide range of languages and systems. FLAVERS has been applied to the analysis of Ada and C++ and Java is underway. It has also been used for the analysis of network protocols and architecture specifications.

Work on the Ada version of FLAVERS had been aimed at continuing to speed the system up and improve the system's handling of more features of the language. A great deal of emphasis was also placed throughout the project upon usability. There was considerable effort devoted to developing a graphical user interface. It resulted in the development of a user interface that

makes FLAVERS far more accessible to wider classes of non-expert users. This work was intended to facilitate the transition of this technology to foster adoption and the consequent ability to evaluate this technology more effectively and thoroughly.

During the latter stages of the project we also developed a process-driven user interface to FLAVERS. This work used the Little-JIL process programming language and the Juliette interpreter to create a system capable of guiding users through the use of FLAVERS by orchestrating the appearance of appropriate windows from the graphical user interface, and the graying out of buttons to prevent users from attempting to use FLAVERS in inappropriate ways. This work can be seen as a continuation of long-standing research in how to make integrated toolsets and environments more effective and better integrated. Thus, this EDCS project has made an important contribution to this area of research which is increasingly being understood and appreciated by the practitioner community in view of its growing interest in integrated programming environments.

In the final stages of this project, a great deal of effort was devoted to continuing the development of a Java version of FLAVERS. This work had been initiated earlier in the project, but continued on through prototype system development during the final year. Much activity involved learning about and interfacing to Java language front end services provided by our colleagues at Kansas State University and McGill University. In addition, we continued to develop our Java concurrency models, and created prototype implementations of analysis capabilities for Java. We also worked to converge the internal architectures of our Ada and Java analyzers, moving us strongly in the direction of being able to analyze systems of mixed Ada and Java code.

Software Process Technology:

Little-JIL is a graphical language developed by the University of Massachusetts for defining processes that coordinate the activities of autonomous agents and their use of resources during the performance of a task. Little-JIL programs are executable so that agents can be guided through a process while ensuring that their actions adhere to the process. Little-JIL programs are also statically analyzable to ensure that reliability requirements are satisfied for all executions of the process.

Flexible and adaptive, a Little-JIL process program defines a variety of ways of accomplishing tasks that can work with varying resource requirements and varying agent capabilities. Agents may be human or automated (software or robots, for example). The choice of particular techniques for a particular context can be made automatically based on resource availability or left up to intelligent agents. Thus, Little-JIL process programs need not tightly control the behavior of agents, but rather guide them through the maze of alternatives and facilitate their communication and resource sharing.

Semantically rich, the Little-JIL language provides features that allow proactive control flow as well as the ability to react to error situations and external events. Pre- and post-requisites are used to dynamically verify that the process is being applied correctly. Resources are defined

using a rich resource model and are reserved and locked during the execution of a process. An agenda manager provides communication with the agents using a graphical user interface for human agents and an API for automated agents.

Language Features

- Task-centered semantics that support multi-agent coordination High-level proactive control constructs allow scheduling and drive execution forward
- Reactions support event-driven processes
- Powerful exception handling for recovery from failures
- Pre and post-requisites help to detect and manage process deviations
- Resource modelling and management guides and constrains execution
- Information flow represents communication between tasks
- Visual notation facilitates understandability and conciseness

On this project we spent considerable effort on improving the Little-JIL language and the Juliette interpreter for Little-JIL. These activities were strongly motivated by considerable experimentation with the application of the language to diverse types of process areas. We wrote Little-JIL process programs in such areas as electronic commerce, data mining, robotics, and intelligent vehicles. Our electronic commerce processes centered on processes for supervising auctions. It is particularly noteworthy that this work entailed collaboration with the finite state verification work, in that the FLAVERS data flow analysis system was used to analyze auction processes written in Little-JIL. This was an important demonstration of the way in which process programs can be analyzed for adherence to key process properties.

As a result of these experimental applications of Little-JIL we made some significant improvements to the language. We clarified our notion of “resource bounded iteration”, an idiom that had been in common use since the language was first conceptualized, adding small but important improvements to the specification of child steps, and clarifying the semantics of our exception management.

Our research on this project also emphasized investigation of the problem of specifying resources. We developed an initial prototype resource specification and management system. Our early experiences with it indicated several shortcomings of this system. As a consequence we designed and implemented an improved version of this system. This early research suggests that resource specification and management is a difficult problem, with great importance to process definition and execution, but it is a problem that has not received adequate attention from the research community.

On this project we also completed the implementation of an early version of the Juliette interpreter. We demonstrated this system at the International Conference on Software Engineering. This demonstration was centered on showing how Little-JIL and Juliette could be used to organize the FLAVERS toolset. This project was referred to in the previous section. From the perspective of the software process technology effort this project should be viewed as a demonstration of how software process technology has matured to the point where it can be used

to implement a process-centered environment. This environment addresses a portion of the software verification phase of development.

Self Adaptive Software:

On this project we also worked on robot team coordination. Much of the work was aimed at understanding the notion of a containment unit. These software entities in this project have the important role of assuming responsibility for specific forms of failure resistance. We designed a number of containment units, and began implementation of some of them. These implementation efforts led to more specific understandings of the nature and role of containment units. This in turn improved our understanding of the nature of self-adaptation. One of the significant outcomes of this project has been the identification of the importance and difficulty of self-adaptation in software. It appears that our process and verification technologies can have considerable importance in studying this problem area.

Distributed Object Technology:

Another key accomplishment of this project was the development of our Grapevine agenda management system, concluding with the completion of a Ph.D. thesis on this topic. The system was developed as a prototype. Evaluation of this prototype was by means of its application to the generation of an agenda management system for the coordination of the components of the Juliette interpreter, and a system for coordination of a laboratory meeting scheduling. These evaluations strongly indicate that further development and refinement of our agenda management ideas is indicated.

DISSEMINATION, PUBLICATION, AND STUDENT TRAINING:

This section summarizes activities in these areas.

During this project our personnel wrote many papers and technical reports listed below. We are gratified that many of these appeared in the most prestigious publication venues, such as the International Conference on Software Engineering and the Foundations of Software Engineering Conference.

In addition, project personnel gave numerous presentations around the country and the world. A partial listing is given here.

Finally we note that PhD students, Eric McCall and Gleb Naumovich completed their Ph.D. degrees through support of this project, and that several Masters students also completed their degrees.

Publications:

Applying Static Analysis to Software Architectures, Gleb Naumovich, George S. Avrunin, Lori A. Clarke and Leon J. Osterweil, Proceedings of Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE 5) and the Sixth European Software Engineering Conference (ESEC 1997), September 1997, Zurich, Switzerland, pp. 77-93.

The Design of a Next-Generation Process Language, Stanley M. Sutton, Jr. and Leon J. Osterweil, Proceedings of Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE 5) and the Sixth European Software Engineering Conference (ESEC 1997), September 1997, Zurich, Switzerland, pp. 142-158.

An Anytime Approach to Analyzing Software Systems, Dan Rubenstein, Leon Osterweil and Shlomo Zilberstein, Proceedings of the 10th International FLAIRS Conference (Florida Artificial Intelligence Research Society), May 12-14, 1997, Daytona Beach, Florida, pp. 386-91.

Software Processes Are Software Too, Revisited: An Invited Talk on the Most Influential Paper of ICSE 9, Leon J. Osterweil, Proceedings of the 19th Nineteenth International Conference on Software Engineering, May 1997, Boston, MA, pp. 540--548.

The Criticality of Modeling Formalisms in Software Design Method Comparison, Rodion M. Podorozhny and Leon J. Osterweil, Proceedings of the 19th International Conference on Software Engineering, May 17-23 1997, Boston, MA, pp. 303-313.

Verification of Concurrent Software with FLAVERS, Gleb Naumovich, Lori A. Clarke, Leon J. Osterweil and Matthew B. Dwyer, Research Demonstration at the 19th International Conference on Software Engineering, May 17-23, 1997, Boston, MA, pp. 594-595.

Gaining Confidence in Distributed Systems, Gleb Naumovich, Lori A. Clarke, Leon J. Osterweil and Matthew B., Dwyer, Proceedings of the 10th International Software Quality Week, Vol. II, Session 10, May 27-30, 1997, San Francisco, CA.

An Empirical Comparison of Static Concurrency Analysis Techniques, A. T. Chamillard, Lori A. Clarke and George S. Avrunin, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 96-84, (Revised: May 1997).

An Adaptable Generation Approach to Agenda Management, Eric K. McCall, Lori A. Clarke and Leon J. Osterweil, Proceedings of the 20th International Conference on Software Engineering (ICSE 1998), April 1998, Kyoto, Japan, pp. 282-291.

Consistency Management for Complex Applications, Peri Tarr and Lori A. Clarke, Proceedings of the 20th International Conference on Software Engineering (ICSE 1998), April 1998, Kyoto, Japan, pp. 230-239.

An Evaluation of Object Management Systems Architectures for Software Engineering Applications, Jayavel Shanmugasundaram, Barbara Staudt Lerner and

Lori A. Clarke, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 97-47.

The Design of a Next-Generation Process Language, Stanley M. Sutton, Jr. and Leon J. Osterweil, Proceedings of the Combined Sixth European Software Engineering Conference and Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering, September 1997, Zurich, Switzerland. Appears in Springer Verlag Lecture Notes in Computer Science #1301, pp. 142-158.

Applying Static Analysis to Software Architectures, Gleb Naumovich, George S. Avrunin, Lori A. Clarke and Leon J. Osterweil, Proceedings of the Combined Sixth European Software Engineering Conference and Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering, Zurich, September 1997, Switzerland. Appears in Springer Verlag Lecture Notes in Computer Science #1301, pp. 77-93.

Experience Using the JIL Process Programming Language to Specify Design Processes, Stanley M. Sutton, Jr., Barbara Staudt Lerner and Leon J. Osterweil, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 97-68.

Programming Parallel Workflows in JIL, S. Sutton, paper presented at IASTED, March 1998, Washington, DC.

Verifying Properties of Distributed Systems: Prospects for Practicality, Lori A. Clarke and Leon J. Osterweil, Proceedings of the Fourth International Conference on Achieving Quality in Software: Software Quality in the Communication Society (AQuiS), March 30-April 2 1998, Venice, Italy, pp. 11-16.

Representing and Reasoning about Knowledge Discovery Processes, David Jensen, Yulin Dong, Barbara Staudt Lerner, Leon J. Osterweil, Stanley Sutton Jr., and Alexander E. Wise, submission to the Fourth International Conference on Knowledge Discovery and Data Mining, August 27-31, 1998, New York, NY.

Enhancing Design Methods to Support Real Design Processes, Barbara Staudt Lerner, Stanley Sutton, Jr., Leon J. Osterweil, Computer Science Department, University of Massachusetts, Amherst, Technical Report 98-06, February 1998. A condensed version of this paper appears in the Proceedings of the 9th International Workshop in Software Specification and Design (IWSSD 9), April 1998 Ise-shima, Japan.

Data Flow Analysis for Checking Properties of Concurrent Java Programs, Gleb Naumovich, George S. Avrunin, and Lori A. Clarke. Proceedings of the 21st International Conference on Software Engineering (ICSE 1999), May 1999, Los Angeles, CA, pp. 399-410.

A Conservative Data Flow Algorithm for Detecting All Pairs of Statements that May Happen in Parallel, Gleb Naumovich and George S. Avrunin, Proceedings of the 6TH ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE 6), November 1998, Lake Buena Vista, FL, pp. 24-34.

Enhancing Design Methods to Support Real Design Processes, Barbara Staudt Lerner and Stanley M. Sutton Jr., Computer Science Department, University of Massachusetts, Amherst, February 1998. A condensed version of this paper appears in the Proceedings of the Ninth IEEE International Workshop on Software Specification and Design, April 16-18, 1998, Ise-shima, Japan.

An Adaptable Generation Approach to Agenda Management, Eric K. McCall, Lori A. Clarke, and Leon J. Osterweil, Proceedings of the 20th International Conference on Software Engineering, Kyoto, Japan, April 1998, pp. 282-291.

Consistency Management for Complex Applications, Peri L. Tarr and Lori A. Clarke. Proceedings of the 20th International Conference on Software Engineering, Kyoto, Japan, April 1998, pp. 230-239.

Comparing Implementation Strategies for Composite Data Flow Analysis Problems, Gleb Naumovich, Lori A. Clarke and Leon J. Osterweil, Proceedings of SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering, June 1998.

Specifying Coordination in Processes Using Little-JIL, Leon Osterweil, Barbara Staudt Lerner, Stan Sutton, Sandy Wise, and Eric McCall, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 98-38.

Multi-Dimensional Separation of Concerns, Stan Sutton, co-authored with Peri Tarr, Harold Ossher, and William Harrison, IBM's T. J. Watson Research Center, Report 21452, April 1999.

Improving Architectural Description Languages to Support Analysis Better, Lori A. Clarke, Proceedings of International Workshop on the Role of Software Architecture in Testing and Analysis (ROSATEA), June 6-July 7, 1998, Marsala, Sicily, Italy, pp. 78-80.

Architecting Processes are Key to Software Quality, Leon Osterweil, Proceedings of International Workshop on the Role of Software Architecture in Testing and Analysis (ROSATEA), June 6-July 7, 1998, Marsala, Sicily, Italy.

Learning Robot Control - Using Control Policies as Abstract Actions, M. Huber, R.A. Grupen, NIPS '98 Workshop: Abstraction and Hierarchy in Reinforcement Learning, December 1998, Breckenridge, CO.

Modeling Resources for Activity Coordination and Scheduling, Rodion Podorozhny, Barbara Staudt Lerner and Leon J. Osterweil, Proceedings of International Conference on Coordination Models and Languages, April 1999, pp. 307-322. Springer Verlag Lecture Notes in Computer Science #1594.

A Hybrid Architecture for Learning Robot Control Tasks, M. Huber, R. A. Grupen, 1999 AAAI Spring Symposium Series: Hybrid Systems and AI: Modeling, Analysis and Control of Discrete + Continuous Systems, Stanford University, CA.

Coordinating Agent Activities in Knowledge Discovery Processes, David Jensen, Yulin Dong, Barbara Staudt Lerner, Eric K. McCall, Leon J. Osterweil, Stanley M. Sutton, Jr. and Alexander Wise. Proceedings of Work Activities Coordination and Collaboration Conference (WACC'99), February 22, 1999, San Francisco, CA, pp. 137-146.

Evaluation of Little-JIL 1.0 with ISPW-6 Software Process Example, H. Lee, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 99-33, Mar 4, 1999.

Using Partial Order Techniques to Improve Performance of Data Flow Analysis Based Verification, Gleb Naumovich, Lori A. Clarke and Jamieson M. Cobleigh, Proceedings of the ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering (PASTE 1999), pp. 57-65, September 1999, Toulouse, France.

A Framework for Learning Visual Discrimination, J. H. Piater and R. A. Grupen, Proceedings of the 12th International FLAIRS Conference, 1999, Orlando, FL. AAAI Press.

A Rigorous Approach to Resource Management in Activity Coordination, R. M. Podorozhny, B. Staudt Lerner, and L. J. Osterweil, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 99-12, March 4, 1999.

Geometrical Modeling and Real-Time Vision Applications of Panoramic Annular Lens (PAL) Camera, Zhigang Zhu, Edward M. Riseman and Allen R. Hanson, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 99-11, February, 1999.

Data Flow Analysis for Checking Properties of Concurrent Java Programs, G. Naumovich, G. S. Avrunin, and L. A. Clarke, Proceedings of the 21st International Conference on Software Engineering (ICSE'99), May 16-22, 1999, Los Angeles, CA, pp. 399-410.

Formally Defining Coordination Processes to Support Contract Negotiations, Aaron G. Cass, Hyungwon Lee, Barbara Staudt Lerner and Leon J. Osterweil, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 99-29, June 1, 1999.

Fast Generation of Dynamic and Multi-Resolution 360° Panorama from Video Sequences, Zhigang Zhu, Guangyou Xu, Edward M. Riseman and Allen R. Hanson,

Proceedings of the IEEE International Conference on Multimedia Computing and Systems, June 7-11, 1999, Florence, Italy, Vol. 1., pp. 400-406.

Panoramic EPI Generation and Analysis of Video from a Moving Platform with Vibration, Zhigang Zhu, Guangyou Xu, Xueyin Lin, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 23-25, 1999, Fort Collins, Colorado, Vol. 2, pp. 531-537.

Flow Analysis for Verifying Specifications of Concurrent and Distributed Software, Matthew B. Dwyer and Lori A. Clarke, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 00-52, August 23, 1999.

Verifying Properties of Process Definitions, Jamieson M. Cobleigh, Lori A. Clarke, and Leon J. Osterweil, Proceedings of the ACM Sigsoft 2000 International Symposium on Software Testing and Analysis (ISSTA 2000), August 2000, Portland, OR, pp. 96-101.

Logically Central, Physically Distributed Control in a Process Runtime Environment, Aaron G. Cass, Barbara Staudt Lerner, Eric K. McCall, Leon J. Osterweil, and Alexander Wise, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 99-65, November 11, 1999.

Specifying Coordination in Processes Using Little-JIL, Alexander Wise, Barbara Staudt Lerner, Eric K. McCall, Leon J. Osterweil and Stanley M. Sutton, Jr., Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 99-71, November 1999.

Data Flow Analysis for Checking Properties of Concurrent Java Programs, G. Naumovich, G. S. Avrunin, and L. A. Clarke, Proceedings of the International Conference on Software Engineering (ICSE'99), May 16-22, 1999, Los Angeles, CA, pp. 399-410.

Formally Defining Coordination Processes to Support Contract Negotiation, Aaron G. Cass, Hyungwon Lee, Barbara Staudt Lerner, and Leon J. Osterweil, Computer Science Department, University of Massachusetts, June 1999, Amherst, MA, Technical Report 99-30.

Fast Generation of Dynamic and Multi-Resolution 360° Panorama from Video Sequences, Zhigang Zhu, Guangyou Xu, Edward M. Riseman and Allen R. Hanson, Proceedings of the IEEE International Conference on Multimedia Computing and Systems, June 7-11, 1999, Florence, Italy, Vol.1, pp. 400-406.

EPI Generation and Analysis of Video from a Moving Platform with Vibration, Zhigang Zhu, Guangyou Xu, Xueyin Lin, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 23-25, 1999, Fort Collins, Colorado, Vol. 2, pp. 531-537.

HI-PLAN and Little-JIL: a Study of Contrast between Two Process Languages, Hyungwon Lee and Leon J. Osterweil, Proceedings of the International Conference on Software Theory and Practice (ICS 2000) of the 16th IFIP World Computer Congress (9WCC 2000), August 2000, Beijing, PRC, pp. 300-311.

Relocation in Mobile Process-Centered Software Development Environments, Supratik Bhattachryya and Leon J. Osterweil, Proceedings of the Third International Workshop on Mobility in Databases and Distributed Systems (DEXA 2000 Workshops), September 6-8, 2000, Greenwich, UK, pp. 198-203.

Little-JIL/Juliette: A Process Definition Language and Interpreter, Aaron G. Cass, Barbara Staudt Lerner, Eric K. McCall, Leon J. Osterweil, Stanley M. Sutton Jr., and Alexander Wise, Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000), June 4-11, Limerick, Ireland, pp. 754-757.

Frameworks for Reasoning about Agents Based Systems, Leon J. Osterweil and Lori A. Clarke, Proceedings of the 1st Annual Workshop on Infrastructure for Scalable Multi-Agent Systems, June 20, 2000, Barcelona, Spain.

Supporting Communication of Heterogeneous Distributed Agents with Agenda Management Systems, Eric K. McCall, Lori A. Clarke, and Leon J. Osterweil, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 00-32, June 20, 2000.

The Right Algorithm at the Right Time: Comparing Data Flow Analysis Algorithms for Finite State Verification, Jamieson M. Cobleigh, Lori A. Clarke and Leon J. Osterweil. Proceedings of the 23rd International Conference on Software Engineering (ICSE 2001), May 12-19, 2001, Toronto, Canada.

Using Little-JIL to Coordinate Agents in Software Engineering, Alexander Wise, Aaron G. Cass, Barbara Staudt Lerner, Eric K. McCall, Leon J. Osterweil and Stanley M. Sutton, Jr. Proceedings of the Automated Software Engineering (ASE 2000) Conference, September 11-15, 2000, Grenoble, France, pp. 155-163.

Modeling and Managing Resource Utilization in Process, Workflow, and Activity Coordination, Barbara Staudt Lerner, Anoop George Ninan, Leon J. Osterweil and Rodion M. Podorozhny, Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 00-58, August 2000.

Continuous Self-Evaluation for the Self-Improvement of Software, Lori A. Clarke and Leon J. Osterweil. Proceedings of the International Workshop on Self Adaptive Software, (IWSAS), April 2000, Oxford, England, pages 27-39. Springer Verlag Lecture Notes in Computer Science #1926.

Exploiting Hierarchy for Planning and Scheduling, Aaron G. Cass, Krithi Ramamritham and Leon J. Osterweil. Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 00-64, December 2000.

Classifying Properties: An Alternative to The Safety-Liveness Classification, Gleb Naumovich and Lori A. Clarke, Proceedings of the FSE-8 2000 Conference, November 8-10, 2000, San Diego, CA, pp 159-168.

Little-JIL/Juliette: A Process Definition Language and Interpreter, Aaron G. Cass, Barbara Staudt Lerner, Eric K. McCall, and Leon J. Osterweil, Proceedings of the 22nd International Conference on Software Engineering, June 4-11, 2000, Limerick, Ireland, pp 754-757.

An Architecture for Flexible, Evolvable Process-Driven User-Guidance Environments, Timothy J. Sliski, Matthew P. Billmers, Lori A. Clarke and Leon J. Osterweil, Proceedings of the Joint 8th European Software Engineering Conference (ESEC 2001) and the 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE 9), September 2001, Vienna, Austria, pp. 33-43.

Exploiting Hierarchy for Planning and Scheduling, Aaron G. Cass, Krithi Ramamritham and Leon J. Osterweil. Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 00-64, December 2000.

Applying Real-Time Scheduling Techniques to Software Processes: A Position Paper, Aaron G. Cass and Leon J. Osterweil. Proceedings of the Eighth European Workshop on Software Process Technology (EWSPT 2001), June 2001, Witten, Germany, pp. 145-151. Springer Verlag Lecture Notes in Computer Science #2077.

Comparison Process Specification for a Repeatable Comparison of Architecting Processes, Rodion M. Podorozhny, Leon J. Osterweil, DeWayne E. Perry. Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 01-06, February 2001.

Design Guidance Through the Controlled Application of Constraints, Aaron G. Cass and Leon J. Osterweil. Proceedings of the 10th International Workshop in Software Specification and Design (IWSSD 10), November 2000, San Diego, CA, pp. 195-199.

The Right Algorithm at the Right Time: Comparing Data Flow Analysis Algorithms for Finite State Verification, Jamieson M. Cobleigh, Lori A. Clarke and Leon J. Osterweil, Proceedings of the 23rd International Conference on Software Engineering (ICSE 2001), May 2001, Toronto, Canada, pp 37-46.

Extending FLAVERS to Check Properties On Infinite Executions of Concurrent Software Systems, Gleb Naumovich and Lori A. Clarke, Proceedings of the Monterey Workshop 2001 on Engineering Automation for Software Intensive System Integration, June 2001, Monterey, CA pp. 267-276.

Using Containment Units for Self Adaptation Of Software, Barbara Staudt Lerner, Jamieson M. Cobleigh, Lori A. Clarke and Leon J. Osterweil. Computer Science Department, University of Massachusetts, Amherst, MA, Technical Report 01-45, September 2001.

An Architecture for Flexible, Evolvable Process-Driven User Guidance Environments, Timothy J. Sliski, Matthew P. Billmers, Lori A. Clarke and Leon J. Osterweil, Proceedings of the Joint Eighth European Software Engineering Conference (ESEC 2001) and the Ninth ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE 9), September 2001, Vienna, Austria pp. 33-43.

Architecting Dynamic Systems Using Containment Units, Leon J. Osterweil, Alexander Wise, Jamieson M. Cobleigh, Lori A. Clarke and Barbara Staudt Lerner, Proceedings of the Working Conference on Complex and Dynamic Systems Architecture, December 2001, Brisbane, Australia.

PROPEL: An Approach Supporting Property Elucidation, Rachel L. Smith, George S. Avrunin, Lori A. Clarke and Leon J. Osterweil, Proceedings of the 24th International Conference on Software Engineering (ICSE 2002), May 2002, Orlando, FL, pp. 11-21.

FLAVERS: a Finite State Verification Technique for Software Systems, Jamieson M. Cobleigh, Lori A. Clarke and Leon J. Osterweil. IBM Systems Journal on Software Testing and Verification, Vol. 41(1), November 2002, pp. 140-165.
URL: <http://www.research.ibm.com/journal/sj/411/cobleigh.html>